MLF

Raising ML to the Power of System F

Didier Le Botlan - Didier Rémy

INRIA
Two type systems ... 

**ML**

Full type inference

\[ \tau ::= \alpha \mid \tau \to \tau \]

\[ \sigma ::= \tau \mid \forall \alpha \cdot \sigma \]

**MLF**

partially annotated

monotypes

**System F**

Fully annotated

\[ t ::= \alpha \mid t \to t \]

\[ t ::= \forall \alpha \cdot t \]
Ambitions

➢ Type all **ML** programs.
➢ Encode all **System F** programs.
➢ Do not guess polymorphism.
➢ Use type annotations instead.
➢ Polymorphism is propagated
Propagate polymorphism

\[ \lambda \alpha. \lambda x : \alpha. \ x \quad \lambda x. \ x \]
\[ \forall \alpha. \alpha \rightarrow \alpha \quad \forall (\alpha) \alpha \rightarrow \alpha \]
\[ \lambda (x : \sigma_{id}). \ x \ [\sigma_{id}] \ x \quad \lambda (x : \sigma_{id}). \ x \ x \]

let \( x = \triangle \) in \( x \) id

\textbf{test-1 :}
\[ (\lambda x : \sigma_{id} \rightarrow \sigma_{id}. \ x \ id) \ \triangle \quad (\lambda x. \ x \ id) \ \triangle \]

\textbf{step is}
\[ \lambda f^{i \rightarrow i}. \lambda x^i. \ f (x^2) - 1 \quad \lambda f. \lambda x. \ f (x^2) - 1 \]

let \( x = \text{step} \) in \( x \) id

\textbf{test-2 :}
\[ (\lambda x^t. \ x \ (id [i])) \ \text{step} \quad (\lambda x. \ x \ id) \ \text{step} \]
Previous work

- Finite Ranks
- Type Annotations to Work
- Colored Local Type Inference
- Semi-Explicit Polymorphism
- ML$^F$ with annotations
What is the type of $f = \lambda x. x \text{ id}$?

* $f \Delta$ is typable,
  hence $f$ has type $\sigma_{\text{id}} \rightarrow \sigma_{\text{id}}$.

* $f \text{ step}$ is typable,
  hence $f$ has type $((i \rightarrow i) \rightarrow (i \rightarrow i)) \rightarrow i \rightarrow i$. 

Syntax of types
Syntax of types

What is the type of \( f = \lambda x. x \text{id} \)?

* \( f \Delta \) is typable,
  hence \( f \) has type \( \forall (\alpha = \sigma_{\text{id}}) (\alpha \to \alpha) \to \alpha \)

* \( f \text{ step} \) is typable,
  hence \( f \) has type \( \forall (\alpha = \text{i} \to \text{i}) (\alpha \to \alpha) \to \alpha \)
Syntax of types

Let $\sigma$ be the type of $f = \lambda x. x \text{id}$.

$\sigma$ can be instantiated to

$\forall (\alpha = \sigma_{\text{id}}) (\alpha \rightarrow \alpha) \rightarrow \alpha$ and $\forall (\alpha = i \rightarrow i) (\alpha \rightarrow \alpha) \rightarrow \alpha$

We write $\sigma = \forall (\alpha \geq \sigma_{\text{id}}) (\alpha \rightarrow \alpha) \rightarrow \alpha$

$\sigma_{\text{id}}$ is $\forall (\alpha \geq \bot) \alpha \rightarrow \alpha$
Syntax of types

monotypes \( \tau ::= \alpha | \tau \rightarrow \tau' \)

type schemes \( \sigma ::= \tau | \bot | \forall (\alpha \geq \sigma_1) \sigma_2 | \forall (\alpha = \sigma_1) \sigma_2 \)

Remark that \( \alpha > \tau \) and \( \alpha = \tau \) are equivalent.
Instance

Where?
\( \forall (\alpha_1 \geq \sigma_1) \ \forall (\alpha_2 = \sigma_2) \ \sigma_3 \)

\( \square \quad \times \quad \square \)

What?
\( \bot \subseteq \sigma \)

Example
\[ \sigma_{id} = \forall (\alpha \geq \bot) \ \alpha \rightarrow \alpha \]
\[ \square \quad \forall (\alpha \geq i) \ \alpha \rightarrow \alpha \]
\[ \equiv \quad i \rightarrow i \]
ML Typing Rules

Var  Fun  App
Gen  Inst  Let
Prefixes \( Q \) are of the form \( \alpha_1 > \bot, \ldots \alpha_n > \bot \)

In \( \text{MLF} \), prefixes are of the form \( \alpha_1 \diamond \sigma_1, \ldots \alpha_n \diamond \sigma_n \)
- Monotypes and type schemes
- Prefixed-typing rules of $\text{ML}$ and principal types.
- Type Inference algorithm of $\text{ML}$.
- Unification algorithm similar to $\text{ML}$’s.
- Both are sound and complete.